Multiresolution maximum intensity volume rendering by morphological adjunction pyramids

Roerdink, Jos B.T.M.

Published in:
Ieee transactions on image processing

DOI:
10.1109/TIP.2003.812759

Publication date:
2003

Multiresolution Maximum Intensity Volume Rendering by Morphological Adjunction Pyramids

Jos B. T. M. Roerdink, *Member, IEEE*

***Abstract*—We describe a multiresolution extension to maximum intensity projection (MIP) volume rendering, allowing progressive refinement and perfect reconstruction. The method makes use of morphological adjunction pyramids. The pyramidal analysis and synthesis operators are composed of morphological 3-D erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis. In this case the MIP operator can be interchanged with the synthesis operator. This fact is the key to an efficient multiresolution MIP algorithm, because it allows the computation of the maxima along the line of sight on a coarse level, before applying a two-dimensional synthesis operator to perform reconstruction of the projection image to a finer level. For interpolation and resampling of volume data, which is required to deal with arbitrary view directions, morphological sampling is used, an interpolation method well adapted to the nonlinear character of MIP. The structure of the resulting multiresolution rendering algorithm is very similar to wavelet splatting, the main differences being that i) linear summation of voxel values is replaced by maximum computation, and ii) linear wavelet filters are replaced by nonlinear morphological filters.**

***Index Terms*—Maximum intensity projection, morphological adjunction pyramids, multiresolution signal decomposition, volume rendering.**

I. Introduction

VOLUME rendering is a technique to produce two-dimensional images of three-dimensional data from different viewpoints, using advanced computer graphics techniques such as illumination, shading and color. Although computing power is rapidly increasing, interactive rendering of volume data is still a demanding problem due to the sizes of the data sets. Also, the desire to exchange volume data through the internet has created a need for fast and efficient methods of transfer and display. For this purpose multiresolution models are developed, allowing systematic decomposition of the data into versions at different levels of detail, which can be used to visualize data incrementally as it arrives ("progressive refinement"). As long as a user is interacting with the data in *preview mode*, e.g., by continuously changing the view direction, a coarse approximation is used, which can be rendered much faster than the full-size data, thus improving user interaction. When needed, this coarse approximation can be refined to full resolution for close inspection.

A well-known volume rendering method [1] is *X-ray volume rendering*, which is based upon integrating the 3-D data along the line of sight. An extensively studied class of multiresolution models in X-ray volume rendering is based on wavelets [2]–[4]. Recent developments include wavelet splatting [5], [6], which extends splatting [7] by using wavelets as reconstruction filters, and Fourier-wavelet volume rendering [8], [9], a generalization of standard Fourier volume rendering [10] which uses a frequency domain implementation of the wavelet transform. All these methods apply to the case of *linear* volume rendering.

The goal of this paper is to present a multiresolution algorithm for maximum intensity projection (MIP) volume rendering, where one computes the *maximum*, instead of the integral, along the line of sight. Because of its computational simplicity, MIP is widely used in the display of magnetic resonance angiography (MRA) and ultrasound data. Since MIP is a nonlinear transformation, the linear wavelet-based multiresolution methods mentioned above are not applicable. Therefore we introduce here a multiresolution scheme based on *morphological*, i.e., nonlinear, pyramids. Such pyramids were recently introduced by Goutsias and Heijmans [11], [12]. Even though the morphological operators are nonlinear and noninvertible, the pyramid scheme does allow perfect reconstruction as well as progressive refinement, just as in the linear case.

For the case of MIP, the class of *adjunction pyramids* [11] where one uses erosion for pyramid analysis and dilation for pyramid synthesis is particularly appropriate, because in this case the MIP operator can be interchanged with the synthesis operator. This fact is the key to an efficient MIP algorithm, because it allows to compute the maxima along the line of sight on a coarse level (where the size of the data is reduced), before applying a two-dimensional synthesis operator to perform reconstruction of the projection image to a finer level. Also, adjunction pyramids have the property that the global approximation error decreases monotonically as detail signals are added. If the data are of integer type (bytes or shorts, depending on the dynamic range), then in contrast to the case of wavelet-based volume rendering, no floating point computations are required, but all operations are carried out as calculations on integers.

To render approximation or detail data of a given level in the pyramid, a single-scale MIP implementation is required. For this purpose, any existing fast MIP implementation can in principle be used, as long as it can work directly on the data structures used to represent the pyramid. Here we use a very fast implementation, i.e., MIP at warp speed [13], which preprocesses the data to remove noncontributing voxels from the volume. Other fast MIP implementations are based on distance encoding [14], or splatting in sheared object space [15]. In many of these methods, pyramids are used as well, albeit as auxiliary data structures. In contrast, the morphological pyramids used

Manuscript received July 18, 2001; revised February 28, 2003. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Nasser Kehtarnavaz.

The author is with the Institute for Mathematics and Computing Science, University of Groningen, 9700 AV Groningen, The Netherlands (e-mail: roe@cs.rug.nl).

Digital Object Identifier 10.1109/TIP.2003.812759

in this paper are not auxiliary data, but an exact representation of the initial data. After the pyramid has been constructed, the original volume data can be discarded, since the pyramid allows perfect reconstruction of the data. Rendering low-resolution approximations by our multiresolution algorithm therefore provides an additional speed-up as compared to the existing single-scale MIP implementations.

The advantages of morphological pyramids in MIP volume rendering are several. First, it allows enhanced user interaction during preview mode because of accelerated rendering of low resolution approximations. When only approximation data on the highest decomposition level L are used, the amount of data, and therefore the rendering time, is reduced by a factor of 8^L, which explains the main motivation for the use of pyramids. A second reason for applying morphological methods in volume rendering is the feature extraction capability of morphological operators. Morphological methods have a well-established mathematical basis [16], [17] and are widely used in image processing for filtering, segmentation (see [18] for a recent review) and shape analysis. Applications of morphological methods in visualization have so far mostly been restricted to preprocessing of volume data, but this is beginning to change now. For example, Lürig and Ertl [19] used multiscale morphological operators as an alternative to transfer functions in traditional color-opacity volume rendering. Visualization of solids defined by morphological operators was considered in [20]. We believe that the possibility of morphological methods to integrate feature extraction within the volume rendering process has great potential and deserves to be explored further.

In [21] we presented some preliminary results on multiresolution MIP rendering for the case of *axial* projections, where the viewing coordinate system is aligned with the grid of volume data. Here we present a full discussion of the general case of nonaxial projections, which requires reconstruction of a continuous function from discrete data before projection. To that end morphological sampling is used [17], [22], an interpolation method well adapted to the nonlinear character of MIP. Also, in [21] we restricted ourselves to *flat* adjunction pyramids, where minima and maxima are computed in a local neighborhood of each voxel. Flatness in particular means that no new grey values are introduced in the analysis of a signal. Here we also study *nonflat* adjunction pyramids, which are useful when one wants to extract detailed features, such as small veins in angiographic data.

The organization of this paper is as follows. Section II gives a few preliminaries on morphological pyramids. In particular, the class of adjunction pyramids is described. Section III then studies multiresolution maximum intensity projection based on morphological adjunction pyramids, and a multiresolution MIP rendering algorithm (MMIP) allowing progressive refinement is derived. Both flat and nonflat adjunction pyramids are considered. Experimental results are given in Section IV. Section V contains a summary and discussion of future work.

II. Morphological Pyramids

A. Multiresolution Signal Decomposition

In multiresolution signal decomposition [11], the goal is to decompose the original signal f which is an element of the signal space V_0 into a number of coarser signals f_j, $j = 0, 1, 2, \ldots$. Here j indicates the *level* of the decomposition. It is assumed that the signals $\{f_j\}$ are elements of associated signal spaces V_j, which all have the structure of the complete lattice $\mathrm{Fun}(\mathbb{Z}^d, \mathcal{T})$ of functions on the discrete grid $\mathbb{Z}^d$, taking values in another complete lattice $\mathcal{T}$. Signal *decomposition* or *analysis* proceeds by analysis operators $\psi_j^{\uparrow}: V_j \to V_{j+1}$, which map a signal to a level higher in the pyramid, thereby reducing information. Signal *reconstruction* or *synthesis* proceeds by synthesis operators $\psi_j^{\downarrow}: V_{j+1} \to V_j$, which map a signal to a level lower in the pyramid (lost information is mapped back). To guarantee that information lost during analysis can be recovered in the synthesis phase in a nonredundant way, one needs the so-called pyramid condition. The analysis and synthesis operators $\psi_j^{\uparrow}$, $\psi_j^{\downarrow}$ are said to satisfy the *pyramid condition* if $\psi_j^{\uparrow}\psi_j^{\downarrow}(f) = f$ for all $f \in V_{j+1}$.

Decomposition of a signal $f \in V_0$ proceeds by the recursion

$$\begin{aligned} f_0 &= f \\ f_{j+1} &= \psi_j^{\uparrow}(f_j), \qquad j \geq 0 \\ d_j &= f_j \dot{-} \psi_j^{\downarrow}(f_{j+1}). \end{aligned}$$

A level-L decomposition of a signal f results in a sequence d_0, d_1, …, d_{L-1}, f_L, where $\{d_j\}$ are detail signals and f_L is a signal at the coarsest level. Here $\dot{-}$ is a generalized subtraction operator. Assuming there exists an associated generalized addition operator $\dot{+}$ such that $\hat{f} \dot{+} (f \dot{-} \hat{f}) = f$ if $f \in V_j$ and $\hat{f} = \psi_j^{\downarrow}\psi_j^{\uparrow}(f)$, *perfect reconstruction* holds, that is, $f \in V_0$ can be exactly reconstructed by

$$f_j = \psi_j^{\downarrow}(f_{j+1}) \dot{+} d_j, \qquad j = L-1, L-2, \ldots, 0.$$

The operators $\dot{+}$ and $\dot{-}$ can be ordinary addition and subtraction, but other choices are possible, as will be seen below.

Approximations of a signal $f \in V_j$ are signals in V_j which are reconstructed from higher levels of the pyramid by omitting some of the detail signals. Below, we will always consider approximations in V_0, i.e., signals are reconstructed to original resolution.

B. Adjunction Pyramids

The class of *morphological adjunction pyramids* [11] satisfies the following assumptions:

1. $V_j = \mathrm{Fun}(\mathbb{Z}^d, \mathcal{T})$, the space of grey-value functions, where the grey-value set $\mathcal{T}$ is a complete lattice.

2. The analysis and synthesis operators are the same at each level: $\psi_j^{\uparrow} = \psi^{\uparrow}$, $\psi_j^{\downarrow} = \psi^{\downarrow}$.

3. $\psi^{\uparrow}: V_0 \to V_1$ and $\psi^{\downarrow}: V_1 \to V_0$ form an *adjunction* between V_0 and V_1. This means that $\psi^{\uparrow}$ is an *erosion*, i.e., commutes with infima, and $\psi^{\downarrow}$ is a *dilation*, i.e., commutes with suprema. In particular, the product operator $\psi^{\downarrow}\psi^{\uparrow}$ is an opening.

First, we consider *flat* pyramids, where local minima and maxima are computed in a local neighborhood $A \subseteq \mathbb{Z}^d$ of

each point. Non-flat pyramids are considered below (see Section III-D). For a d-dimensional signal f, the analysis and synthesis operators of a flat adjunction pyramid have the form

$$\psi_A^{\uparrow}(f)(n) = \sigma^{\uparrow}\varepsilon_A(f)(n) = \bigwedge_{k\in A} f(2n+k) \tag{1}$$

$$\psi_A^{\downarrow}(f)(k) = \delta_A\sigma^{\downarrow}(f)(k) = \bigvee_{n\in A[k]} f\left(\frac{k-n}{2}\right). \tag{2}$$

Here, $\delta_A(f)$ and $\varepsilon_A(f)$ are the flat dilation and erosion with structuring element A, whereas $\sigma^{\uparrow}$ and $\sigma^{\downarrow}$ denote downsampling and upsampling by a factor of 2 in each spatial dimension. Also, $\mathbb{Z}^d[n] = \{k \in \mathbb{Z}^d | k - n \in 2\mathbb{Z}^d\}$ and $A[n] = A \cap \mathbb{Z}^d[n]$. The sets $A[n]$ form a disjoint partition of A of at most 2^d nonempty and mutually disjoint subsets. So the analysis phase consists of erosion followed by downsampling; the synthesis phase consists of upsampling followed by dilation. As shown in [11], the pyramid condition is satisfied if there exists an $a \in A$ such that the translates of a over an even number of grid steps are never contained in the structuring element A, that is, when

$$A[a] = \{a\} \qquad \text{for some } a \in A. \tag{3}$$

The product $\psi_A^{\downarrow}\psi_A^{\uparrow}$ is an *opening*, i.e., an operator which is increasing, anti-extensive and idempotent. The anti-extensivity property means that $\psi_A^{\downarrow}\psi_A^{\uparrow}(f) \le f$. In this case, the generalized addition and subtraction operators can be defined by [11]

$$t \dot{+} s = t \vee s \tag{4}$$

$$t \dot{-} s = \begin{cases} t, & \text{if } t > s \\ \bot, & \text{if } t = s \end{cases} \tag{5}$$

where $\bot$ is the least element of $\mathcal{T}$. From now on, we will assume that $\mathcal{T}$ is the set of nonnegative integers, which means that $\bot = 0$. As a consequence, the detail signals are nonnegative:

$$d_j(n) = f_j(n) \dot{-} \psi_A^{\downarrow}(f_{j+1})(n) = f_j(n) \dot{-} \psi_A^{\downarrow}\psi_A^{\uparrow}(f_j)(n) \ge 0. \tag{6}$$

Note that (5) implies that the detail signal $d_j(n)$ equals $f_j(n)$ except at points where $f_j(n)$ equals $\psi_A^{\downarrow}\psi_A^{\uparrow}(f_j)(n)$. So, detail signals are not "small" in regions where the structuring function does not match the data well. Since our primary interest is in the approximations, not the detail signals, this is not a problem, but for compression purposes other choices of addition and subtraction operators may be advantageous.

For an adjunction pyramid with the addition operator defined by (4), the reconstruction takes a special form. Making use of the fact that $\psi_A^{\downarrow}$ is a dilation, hence commutes with suprema, one easily derives

$$f = \psi_A^{\downarrow^L}(f_L) \vee \bigvee_{k=0}^{L-1} \psi_A^{\downarrow^k}(d_k) \tag{7}$$

where L is the decomposition depth and $\psi_A^{\downarrow^k}$ denotes k-fold composition of $\psi_A^{\downarrow}$ with itself. This representation is quite similar to the (linear) Laplacian pyramid representation [23]. The main difference is that sums have been replaced by maxima.

III. Multiresolution Maximum Intensity Projection

Now we address the main problem of this paper, which is the derivation of a multiresolution maximum intensity projection (MMIP) volume rendering algorithm with progressive refinement. To allow multiresolution reconstruction in MIP, morphological adjunction pyramids are used.

For clarity of exposition, we first consider the case of *axial* projections (axes of the viewing coordinate system parallel to the axes of the original grid of volume data). Subsequently, the extension to arbitrary view directions is discussed. This extension requires that the volume data are interpolated and resampled. For this purpose, morphological sampling is used [17], [22], an interpolation method well adapted to the nonlinear character of MIP.

A. Axial Projections

Consider a discrete volume data set f, with values $f(k, l, m)$, where $(k, l, m) \in S \subseteq \mathbb{Z}^3$. Let $\mathcal{M}_{\mathbf{z}}(f)$ denote the maximum projection of voxels parallel to the z-axis

$$\mathcal{M}_{\mathbf{z}}(f)(k, l) = \bigvee_m f(k, l, m).$$

From the pyramid representation (7), and the fact that $\mathcal{M}_{\mathbf{z}}$ evidently distributes over maxima, one gets

$$\mathcal{M}_{\mathbf{z}}(f) = \mathcal{M}_{\mathbf{z}}\left(\psi_A^{\downarrow^L}(f_L)\right) \vee \bigvee_{k=0}^{L-1} \mathcal{M}_{\mathbf{z}}\left(\psi_A^{\downarrow^k}(d_k)\right). \tag{8}$$

Although, in principle, this formula allows us to do multiresolution MIP, this expression is computationally inefficient, because to compute the projections at a certain level of approximation j, one has to reconstruct f_L, d_{L-1}, …, d_j first to full resolution and then apply the maximum operator $\mathcal{M}_{\mathbf{z}}$. What we really want is to compute the maxima along the line of sight on a coarse level, where the size of the data is reduced, before applying a synthesis operator to perform reconstruction to a finer level. This is possible, as is shown next.

1) Computing the Maxima Before Synthesis: According to (2), the synthesis operator $\psi_A^{\downarrow}$ is composed of upsampling, followed by a dilation. Therefore, our problem is to rewrite $\mathcal{M}_{\mathbf{z}}\psi_A^{\downarrow}(f) = \mathcal{M}_{\mathbf{z}}\delta_A\sigma^{\downarrow}(f)$ such that the MIP operator $\mathcal{M}_{\mathbf{z}}$ is "moved to the right." The problem can be split in two parts. First, the projection of a dilated function is considered, then the projection of an upsampled function, and finally the two results are combined.

First of all, both $\mathcal{M}_{\mathbf{z}}$ and δ_A involve the computation of maxima. Therefore, it is obvious that to compute $\mathcal{M}_{\mathbf{z}}\delta_A(f)$, one can first project f along the z-axis, and then dilate the resulting 2-D function by a flat 2-D structuring function whose support $\tilde{A}$ is the projection of A. That is,

$$\mathcal{M}_{\mathbf{z}}\delta_A(f) = \delta_{\tilde{A}}\mathcal{M}_{\mathbf{z}}(f) \tag{9}$$

with $\tilde{A} := \{(k, l) \in \mathbb{Z}^2 | (k, l, m) \in A \text{ for some } m \in \mathbb{Z}\}$. Note that δ_A is a 3-D dilation while $\delta_{\tilde{A}}$ is a 2-D dilation. If a set is identified with its indicator function, we can also write $\tilde{A} = \mathcal{M}_{\mathbf{z}}(A)$. This form will be useful when considering arbitrary view directions and/or grey scale structuring functions.

Next, consider projection of an upsampled function: $\mathcal{M}_{\mathbf{z}}\sigma^{\downarrow}(f)$. Upsampling means inserting zeroes between neighboring voxels in all three spatial dimensions. If the upsampled function is projected, then for those (k, l) which are in the projection of the support of the original function f the outcome will be unaffected, since the inserted zero values never contribute to the maximum (zero being the minimum data value possible). On the other hand, for those (k, l) which are not in the projection of the support of the original function f, projection means computing the maximum of a vertical line of zeroes, which results in a zero at (k, l). Therefore,

$$\mathcal{M}_{\mathbf{z}}\sigma^{\downarrow}(f) = \sigma^{\downarrow}\mathcal{M}_{\mathbf{z}}(f) \tag{10}$$

where $\sigma^{\downarrow}$ on the right-hand side is a 2-D upsampling operator. Note that the dimension of $\sigma^{\downarrow}$ is not explicitly indicated, since this is clear from the dimension of the domain of the function on which it acts. The final step is to combine (9) and (10), yielding

$$\begin{aligned}\mathcal{M}_{\mathbf{z}}\psi_A^{\downarrow}(f) &= \mathcal{M}_{\mathbf{z}}\delta_A\sigma^{\downarrow}(f) = \delta_{\check{A}}\mathcal{M}_{\mathbf{z}}\sigma^{\downarrow}(f)\\ &= \delta_{\check{A}}\sigma^{\downarrow}\mathcal{M}_{\mathbf{z}}(f) = \psi_{\check{A}}^{\downarrow}\mathcal{M}_{\mathbf{z}}(f)\end{aligned} \tag{11}$$

where $\psi_{\check{A}}^{\downarrow} = \delta_{\check{A}}\sigma^{\downarrow}$ is a 2-D synthesis operator of the same form as $\psi_A^{\downarrow}$ except that the structuring element A has been replaced by $\check{A}$. It is evident that a similar formula holds for projection of iterated versions $\psi_A^{\downarrow^k}$, $k > 1$.

As a result of the above analysis, we have proved the main result of this subsection, which is a multiresolution MIP representation for the case of axial projections

$$\mathcal{M}_{\mathbf{z}}(f) = \psi_{\check{A}}^{\downarrow^L}\left(\mathcal{M}_{\mathbf{z}}(f_L)\right) \vee \bigvee_{k=0}^{L-1} \psi_{\check{A}}^{\downarrow^k}\left(\mathcal{M}_{\mathbf{z}}(d_k)\right). \tag{12}$$

This formula allows us to do multiresolution MIP progressively, starting from the coarsest signal f_L, and successively taking the detail signals d_k, $k = L-1, \ldots, 0$ into account as follows. Define a level-L approximation $\hat{\mathcal{M}}_{\mathbf{z}}^{(L)}(f)$ by $\hat{\mathcal{M}}_{\mathbf{z}}^{(L)}(f) = \psi_{\check{A}}^{\downarrow^L}(\mathcal{M}_{\mathbf{z}}(f_L))$, and level-$j$ approximations $\hat{\mathcal{M}}_{\mathbf{z}}^{(j)}(f)$, $j = L-1, \ldots, 0$, by

$$\hat{\mathcal{M}}_{\mathbf{z}}^{(j)}(f) = \psi_{\check{A}}^{\downarrow^L}\left(\mathcal{M}_{\mathbf{z}}(f_L)\right) \vee \bigvee_{k=j}^{L-1} \psi_{\check{A}}^{\downarrow^k}\left(\mathcal{M}_{\mathbf{z}}(d_k)\right). \tag{13}$$

Then, starting from $\hat{\mathcal{M}}_{\mathbf{z}}^{(L)}(f)$, (13) can be computed recursively by

$$\hat{\mathcal{M}}_{\mathbf{z}}^{(j-1)}(f) = \psi_{\check{A}}^{\downarrow^{j-1}}\left(\mathcal{M}_{\mathbf{z}}(d_{j-1})\right) \vee \hat{\mathcal{M}}_{\mathbf{z}}^{(j)}(f). \tag{14}$$

In particular, $\hat{\mathcal{M}}_{\mathbf{z}}^{(0)}(f) = \mathcal{M}_{\mathbf{z}}(f)$, the exact MIP of the original data f.

B. Arbitrary View Directions

A simple method to produce MIP views for an arbitrary projection direction is *voxel projection*. All voxels are projected in arbitrary order, and each voxel v contributes only to the pixel p which is closest to the projection of the center of v on the view plane. The final value of pixel p is the maximum of the values of all voxels which project to p. A problem with this method is that holes may appear in the projection image for nonaxial views. Holes are isolated pixels in the image plane to which no voxel projects because of undersampling. To deal with this problem, a continuous function has first to be reconstructed from the discrete data. The standard solution is to use some form of linear interpolation, but this does not combine well with the process of maximum computation. Instead, morphological sampling is used.

Let the direction of projection be defined by a unit vector $\boldsymbol{\theta}$, and let $\mathbf{u}$ and $\mathbf{v}$ be two mutually orthogonal unit vectors in the view plane perpendicular to $\boldsymbol{\theta}$. To fix the viewing coordinate system completely, two angles θ and ϕ are needed for defining the view vector $\boldsymbol{\theta}$, and a third angle α for defining the orientation of the view plane with respect to the view vector $\boldsymbol{\theta}$. The three angles defining the viewing coordinate system will be collectively denoted by $\boldsymbol{\Theta} = (\theta, \phi, \alpha)$.

1) Single-Scale Projections: We choose one of the possible morphological sampling strategies described in [17], [22], by reconstructing a continuous function from discrete data by dilation with a structuring element $K \subseteq \mathbb{R}^3$. Then the reconstructed function is projected on the view plane. Finally, we sample the projected function on a Cartesian pixel grid in the view plane by erosion. It is easy to see that in this way the MIP projection process is split in two phases: i) voxel projection with maximum accumulation, and ii) a final 2-D closing with a structuring element B, where B is the set of pixels which are hit by the projection of K on the view plane. This is analogous to the two-stage approach [24] developed for the so-called splatting method, which is a linear volume rendering technique based on the X-ray transform. In the first phase, all voxels in the volume data set $S \subseteq \mathbb{Z}^3$ are projected on the view plane $P \subseteq \mathbb{Z}^2$, such that a voxel with center (k, l, m) contributes to a pixel (i, j) if the projection of (k, l, m), which is denoted by $\pi_{\boldsymbol{\Theta}}(k, l, m)$, falls within a square $C(i, j)$ of size Δ_u by Δ_v with center (i, j), where Δ_u and Δ_v are the horizontal and vertical sampling distances in the view plane. We will always take $\Delta_u = \Delta_v = \Delta$, where Δ equals the sampling distance of the original volume data, which are assumed to be sampled on a uniform Cartesian grid. Without loss of generality, we set $\Delta = 1$. The final pixel value is the maximum of the values of all voxels which project to the same pixel. In the second phase, a 2-D closing is performed on the image produced in phase 1.

The resulting algorithm can be summarized as follows. The combined operator, which maps the discrete 3-D array f of volume data to a discrete 2-D array of pixel values, is denoted by $\mathbf{M}_{\boldsymbol{\Theta}}$.

1) Voxel projection with maximum accumulation

$$\begin{aligned}&\mathcal{M}_{\boldsymbol{\Theta}}(f)(i, j)\\ &\quad= \bigvee_{(k, l, m)\in S} \{f(k, l, m): \pi_{\boldsymbol{\Theta}}(k, l, m) \in C(i, j)\}.\end{aligned} \tag{15}$$

2) Two-dimensional closing ϕ_B of $\mathcal{M}_{\boldsymbol{\Theta}}(f)$ with structuring element B

$$\mathbf{M}_{\boldsymbol{\Theta}}(f)(i, j) = \phi_B(\mathcal{M}_{\boldsymbol{\Theta}}(f))(i, j), \qquad (i, j) \in P.$$

The structuring element B depends on the 3-D structuring element K used in the morphological interpolation. To minimize the amount of flattening of the projected image due to the closing, K should be as small as possible. On the other hand, K should be large enough to cover the complete volume by translation over integer steps. One obvious choice is to take

K equal to a unit cube surrounding each voxel, so that the translates of K over the voxel lattice fill the volume without overlap. In that case the projected structuring element B becomes dependent upon the view direction. A simpler choice is to take for B a fixed structuring element of size 2×2, except for axial projections where no closing is necessary.

2) Multiscale Projections: Starting point is again the discrete pyramid representation (7). The MIP operator $\mathbf{M}_{\mathbf{\Theta}}$ is composed of the voxel projection operator $\mathcal{M}_{\mathbf{\Theta}}$, and a closing ϕ_B. Clearly, $\mathcal{M}_{\mathbf{\Theta}}$ distributes over maxima, so

$$\mathbf{M}_{\mathbf{\Theta}}(f) = \phi_B \left(\mathcal{M}_{\mathbf{\Theta}} \left(\psi_A^{\downarrow^L}(f_L) \right) \vee \bigvee_{k=0}^{L-1} \mathcal{M}_{\mathbf{\Theta}} \left(\psi_A^{\downarrow^k}(d_k) \right) \right). \tag{16}$$

The analysis in Section III-A leading to the commutativity relation (11) is easily generalized to arbitrary directions if one considers MIP projections of a continuous function f, with $\psi_A^{\downarrow}$ consisting of upsampling by a continuous factor followed by a dilation with a continuous structuring element A. Assuming this relation to hold in the discrete case as well, $\mathcal{M}_{\mathbf{\Theta}}$ can be moved inside the $\psi_A^{\downarrow}$ operators, yielding

$$\mathbf{M}_{\mathbf{\Theta}}(f) = \phi_B \left(\psi_{\tilde{A}}^{\downarrow^L} \left(\mathcal{M}_{\mathbf{\Theta}}(f_L) \right) \vee \bigvee_{k=0}^{L-1} \psi_{\tilde{A}}^{\downarrow^k} \left(\mathcal{M}_{\mathbf{\Theta}}(d_k) \right) \right) \tag{17}$$

where again $\tilde{A} := \mathcal{M}_{\mathbf{\Theta}}(A)$ is the projection of the structuring element A on the view plane (cf. Section III-A). This step involves a certain discretization error, which is assessed below experimentally (cf. Section IV-A).

Again we aim for progressive computation of (17). This requires that the additional closing ϕ_B in (17) can be handled "recursively." This is possible in view of the following property.

Lemma: Let (ϵ, δ) be an adjunction on a lattice $\mathcal{L}$. Then the closing $\phi = \epsilon\delta$ satisfies

$$\phi(f \vee g) = \phi(\phi(f) \vee g) = \phi(f \vee \phi(g)) = \phi(\phi(f) \vee \phi(g)).$$

Proof: We prove only the first equality, the proof of the other two is similar. Using the fact that δ commutes with suprema and that $\delta = \delta\epsilon\delta$ (see [17]), one finds

$$\begin{aligned} \phi(f \vee g) &= \epsilon\delta(f \vee g) = \epsilon(\delta(f) \vee \delta(g)) = \epsilon(\delta\epsilon\delta(f) \vee \delta(g)) \\ &= \epsilon(\delta\phi(f) \vee \delta(g)) = \epsilon\delta(\phi(f) \vee g) = \phi(\phi(f) \vee g). \end{aligned}$$

■

Define a level-L approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f)$ by $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f) = \phi_B \left(\psi_{\tilde{A}}^{\downarrow^L} (\mathcal{M}_{\mathbf{\Theta}}(f_L)) \right)$, and level-$j$ approximations $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f)$, $j = L-1, \ldots, 0$, by

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f) = \phi_B \left(\psi_{\tilde{A}}^{\downarrow^L} \left(\mathcal{M}_{\mathbf{\Theta}}(f_L) \right) \vee \bigvee_{k=j}^{L-1} \psi_{\tilde{A}}^{\downarrow^k} \left(\mathcal{M}_{\mathbf{\Theta}}(d_k) \right) \right). \tag{18}$$

Then, using the Lemma, (18) can be computed recursively by

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j-1)}(f) = \phi_B \left(\psi_{\tilde{A}}^{\downarrow^{j-1}} (\mathcal{M}_{\mathbf{\Theta}}(d_{j-1})) \vee \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f) \right). \tag{19}$$

In the case that the sampling distance in the view plane is equal to the sampling distance of the original volume data (as is the case in all the experiments reported below), holes in the projected images will only arise if the view vector is along one of the grid axes of the volume data, but the view plane is rotated around the view vector. We have found that even when the closing ϕ_B is not necessary to prevent holes, it still diminishes the discretization error. Therefore the closing is applied for all nonaxial views.

C. MMIP Algorithm

The proposed multiresolution MIP algorithm can be summarized as follows.

1) *Preprocessing.* Compute an L-level 3-D morphological adjunction pyramid with structuring element A of the volume data, resulting in a sequence $d_0, d_1, \ldots, d_{L-1}, f_L$, where $\{d_j\}$ are detail signals and f_L is a signal at the coarsest level.

2) *Actual MIP Volume Rendering.* For a given orientation $\mathbf{\Theta}$ of the viewing coordinate system, do:

• Compute a low resolution approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f)$ by first applying the voxel projection operator $\mathcal{M}_{\mathbf{\Theta}}$ to f_L, followed by the 2-D synthesis operator $\psi_{\tilde{A}}^{\downarrow^L}$ and a final closing ϕ_B

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f) = \phi_B \left(\psi_{\tilde{A}}^{\downarrow^L} (\mathcal{M}_{\mathbf{\Theta}}(f_L)) \right).$$

Here, $\tilde{A} = \mathcal{M}_{\mathbf{\Theta}}(A)$ is the voxel projection of the 3-D structuring element A, and B is a 2-D structuring element (for axial views, ϕ_B can be omitted).

• Refine the image progressively by taking the detail signals $d_k, k = L-1, \ldots, 0$ into account. From a level-j approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f)$, compute an approximation $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j-1)}(f)$ on level $j-1$ by projecting d_{j-1}, applying the 2-D pyramid synthesis operator $\psi_{\tilde{A}}^{\downarrow^{j-1}}$ to the projection, taking the maximum of the image so obtained with the previous approximation, and finally applying a closing ϕ_B

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j-1)}(f) = \phi_B \left(\psi_{\tilde{A}}^{\downarrow^{j-1}} (\mathcal{M}_{\mathbf{\Theta}}(d_{j-1})) \vee \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f) \right). \tag{20}$$

• The recursion terminates with $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(0)}(f)$, which equals the MIP $\mathbf{M}_{\mathbf{\Theta}}(f)$ of the original data f (up to a discretization error, which is zero for axial projections).

The structure of this algorithm is very similar to that of wavelet splatting [5], [6], [9]. The main differences are that i) linear summation of voxel values is replaced by maximum computation and ii) linear wavelet filters are replaced by nonlinear morphological filters.

1) Error Decrease: From (19) one immediately deduces that $\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f) \leq \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j-1)}(f)$, since from (6) the details signals d_{j-1} are nonnegative. So the projections increase pointwise as one goes down the pyramid

$$\hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L)}(f)(x) \leq \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(L-1)}(f)(x) \leq \cdots \leq \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(0)}(f)(x). \tag{21}$$

Define a global approximation error by $\mathcal{E}^{(j)} = \|\mathbf{M}_{\mathbf{\Theta}}(f) - \hat{\mathbf{M}}_{\mathbf{\Theta}}^{(j)}(f)\|$, where $\|\cdots\|$ is some error norm, such as the L_1, L_2 or L_∞ norm. Then (21) implies that the global approximation error decreases monotonically with decreasing j. An illustration is given in Section IV; see in particular Table I.

TABLE I
DATA SIZES (VALUE-SORTED ARRAY FORMAT) AND RENDERING TIMES OF MIP (FULL IMAGE) AND MMIP (PROGRESSIVE RENDERINGS OF APPROXIMATION AND DETAIL DATA). "ERROR" DENOTES THE RELATIVE L_1-ERROR BETWEEN APPROXIMATION IMAGE AND FULL IMAGE

MRA data $256 \times 256 \times 256$	size (kbytes)	time (s)	error (%)
full image	838.5	0.27	
level 2 approximation	0.85	0.02	0.97
add detail level 1	30.2	0.03	0.63
add detail level 0	801.6	0.26	0.0
CT data $256 \times 256 \times 256$	size (kbytes)	time (s)	error (%)
full image	19715	6.64	
level 2 approximation	277	0.13	0.36
add detail level 1	2052	0.70	0.12
add detail level 0	17231	5.86	0.0

D. Grey Scale Structuring Functions

Let the flat pyramid operators (1) and (2) be replaced by an analysis operator $\psi_a^{\uparrow}$ and synthesis operator $\psi_a^{\downarrow}$ involving erosion and dilation by a structuring function $a \in \mathrm{Fun}(\mathbb{Z}^d, \mathcal{T})$

$$\psi_a^{\uparrow}(f) = \sigma^{\uparrow}\varepsilon_a(f), \qquad \psi_a^{\downarrow}(f) = \delta_a\sigma^{\downarrow}(f)$$

where $(\varepsilon_a, \delta_a)$ is a grey-scale adjunction

$$\delta_a(f)(x) = \bigvee_{y\in A,\, x-y\in F} f(x-y) + a(y),$$
$$\varepsilon_a(f)(x) = \bigwedge_{y\in A,\, x+y\in F} f(x+y) - a(y).$$

Here, A and F denote the domain of a and f, respectively. Then, as shown in [11], the pyramid condition is still satisfied as long as (3) holds. Since in our case $\mathcal{T}$ is the lattice of nonnegative integers, we must ensure that the least element of the grey-scale domain (i.e., the value zero) is conserved by the pyramid operators. This is the case if the structuring function a satisfies the requirement $\mathbf{0} \in A$, $a(y) \leq a(\mathbf{0}) = 0$ for all $y \in A$, where $\mathbf{0}$ is the origin of $\mathbb{Z}^d$. Under this condition, in the analysis step $\psi_a^{\uparrow}(f)$ will always be nonnegative since f itself is nonnegative. In the synthesis step the same holds: after the upsampling operation $\sigma^{\downarrow}$ the value at each point x is still nonnegative, and the final dilation δ_a does not change this. The reason is that, since $\mathbf{0} \in A$ and $a(\mathbf{0}) = 0$, the point x contributes its current—nonnegative—value during the local maximum computation around x. Therefore the outcome of $\psi_a^{\downarrow}(f)(x)$ can never be smaller than zero. It is easy to show that in the case of nonflat structuring functions, the basic commutativity relation (11) still holds if the structuring element A is replaced by the structuring function a, and $\tilde{A}$ by $\tilde{a} = \mathcal{M}_{\mathbf{z}}(a)$. For nonaxial views, we take $\tilde{a} = \mathbf{M}_{\mathbf{\Theta}}(a)$, which again will involve a discretization error. For an example of the use of nonflat structuring functions, see Fig. 3.

E. Time and Memory Requirements

1) Time Complexity: For MIP rendering, the time complexity is $\mathcal{O}(N^3)$ for a volume data set of size N^3, since for each of the N^2 pixels in the image plane, the maximum of N voxel values has to be computed. Of course, various acceleration schemes are possible, but these do not change the complexity [13]–[15]. For MMIP, there is first the preprocessing step, which has only to be executed once. The analysis operator involves 3-D erosions which are linear in the number of voxels, so preprocessing is $\mathcal{O}(N^3)$. A complete reconstruction is $\mathcal{O}(N^3)$ as well, since it is dominated by the projections of the volume data [the 2-D synthesis operators $\psi_{\tilde{A}}^{\downarrow^j}$ and closing ϕ_B are $\mathcal{O}(N^2)$]. But in preview mode, when only approximation data on the highest decomposition level L are used, the amount of data, and therefore the rendering time, is reduced by a factor of 8^L, which explains the main motivation for the use of pyramids.

2) Memory Usage: For a volume data set of size N^3, the size of an L-level analysis pyramid is

$$N^3\left(1 + \tfrac{1}{8} + \cdots + \left(\tfrac{1}{8}\right)^L\right) = \tfrac{8}{7}N^3\left(1 - (1/8)^{L+1}\right) \leq \tfrac{8}{7}N^3.$$

Hence, the pyramid takes at most 14% more memory than the input data. In principle, if morphological wavelet pyramids [12] are used, then this memory overhead can be reduced completely, but this requires the construction of new wavelets adapted to the special choice (4) of the addition/subtraction operations.

The MIP projections $\mathcal{M}_{\mathbf{\Theta}}$ required in the MMIP algorithm can be implemented by means of the object order voxel projection method of Mroz *et al.* [13]. In this method, one loops through the volume, projects all voxels to the image plane with each voxel contributing to exactly one pixel, and accumulates values at pixel locations by maximum computation. The final result is independent of the order in which the voxels are visited. This method also uses an efficient volume data storage scheme, by histogram-based sorting of "interesting" voxels according to grey value, and storing these in a value-sorted array of voxel positions. An additional array contains the cumulative histogram values. In the experiments to be discussed in Section IV, all levels of the pyramid were created and stored as value-sorted arrays. We define interesting voxels simply as those with a nonzero grey value (zero voxel values never contribute to pixel maxima). In practice, especially for angiographic data, a substantial reduction (sometimes more than 95%) in the amount of voxels to be processed is thus obtained. Further memory reductions are possible by using a preprocessing scheme to identify and remove other types of noncontributing voxels [13], and by applying compression techniques (thresholding, Huffman coding, etc.). For this so-called compression domain rendering in the case of wavelet splatting, see [6].

IV. EXPERIMENTAL RESULTS

Multiscale MIP rendering experiments were carried out with a CT head data set and an MR angiography (MRA) data set, both of size 256^3, using a pyramid with two decomposition levels ($L = 2$). We used a PC with a 1.9 GHz Pentium 4 processor and 512 Mb memory. In the pyramid operations, dilations and erosions with a $2 \times 2 \times 2$ structuring function were used. The sampling distance in the view plane was taken equal to the sampling distance of the original volume data. For the CT data,

about 29% of the data consisted of nonzero voxels; for the angiography data, this was 1.25%.

A. Multiscale Discretization Error

To give an indication of the multiscale discretization error when using the approximate equation (17), we computed the difference between $\mathbf{M}_{\boldsymbol{\Theta}}(f)$, the direct MIP projection of the original data, and $\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(0)}(f)$, obtained by the recursion (19). The relative L_1-error, defined as the L_1-norm of this difference divided by the L_1-norm of $\mathbf{M}_{\boldsymbol{\Theta}}(f)$, was computed as a function of angle for the angiographic data set. For nonaxial projections a closing ϕ_B was used with B a 2×2 structuring element. The results of the error computations are shown in Fig. 1. The left error curve, showing results for α varying between 0 and π, is periodic with period $\pi/2$, since the uniform rectangular coordinate grid, which is rotating around the fixed projection vector $\boldsymbol{\theta}$, is invariant under rotation over $\pi/2$. The error is indeed zero for axial projections, i.e., at multiples of $\pi/2$. The maximum error remains below 1%, which is not noticeable in the MIP images. Also, we found no visually disturbing artifacts to occur.

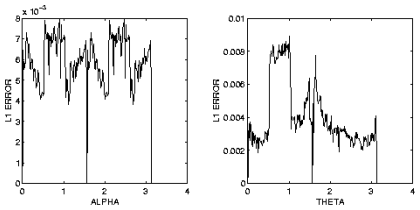

Fig. 1. Multiscale discretization error $\|\hat{\mathbf{M}}_{\boldsymbol{\Theta}}^{(0)}(f) - \mathbf{M}_{\boldsymbol{\Theta}}(f)\|_1 / \|\mathbf{M}_{\boldsymbol{\Theta}}(f)\|_1$ of the MRA data set as a function of rotation angle. Left: $\theta = \phi = 0, \alpha \in [0, \pi]$; right: $\phi = \alpha = 0, \theta \in [0, \pi]$.

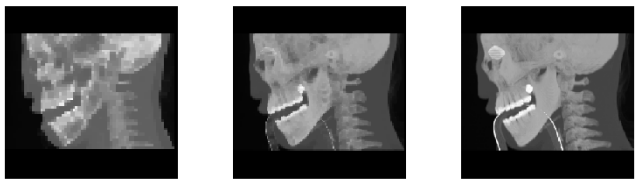

Fig. 2. MMIP reconstruction of the CT data set from a 2-level morphological adjunction pyramid using a flat structuring function with a support of size $2 \times 2 \times 2$. From left to right: approximations on level 2, level 1 and level 0 (original).

B. Performance

Both for the CT and MRA data sets, about 5 s was needed to create the pyramid. To remove some uniform background noise, voxels with value below a small threshold (4% of the maximum grey value) were first set to zero. Sizes in value-sorted array format (see Section III-E) and rendering times (averages over 50 runs) of the successive levels of the pyramid are given in Table I. For comparison, the numbers for direct MIP rendering of the full-size volume data are given as well. All times are excluding I/O. The table also shows the relative L_1-error between a level-j approximation and the full image, for $j = 0, 1, 2$. The timings show that computing a level-2 or level-1 approximation takes considerably less time than a full-size MIP, especially for data sets with a relatively large number of nonzero voxels. Rendering times were found to be almost independent of view angles. Fig. 2 shows successive approximations for the CT data. Here, and in all cases below, axial projection was used.

C. Nonflat Adjunction Pyramids

Higher level approximations quickly remove details of the data, since they are essentially morphological openings by a structuring element whose size increases with level. Note in particular for the head CT data in Fig. 2 that small details such as the tube from the mouth almost disappear in the level 1 approximation. To be useful for angiographic data, the method has to be adapted so that small details are better preserved in higher levels of the pyramid. One possibility is to use nonflat structuring functions, which can probe fine details more accurately. As an example, we computed MIP projections of the MRA data, both with a flat and with a nonflat structuring function with a support of size $2 \times 2 \times 2$. In the nonflat case, the structuring function had a peak of size 25. The results are shown in Fig. 3. For enhanced display purposes, we show the images in reverse-video mode (high intensity corresponding to low grey value). Clearly, the level-1 result where a nonflat structuring function is used contains more small details compared to the case with a flat structuring function, but in level 2 many details still have disappeared. To further improve on this, other types of pyramids are needed, see Section V.

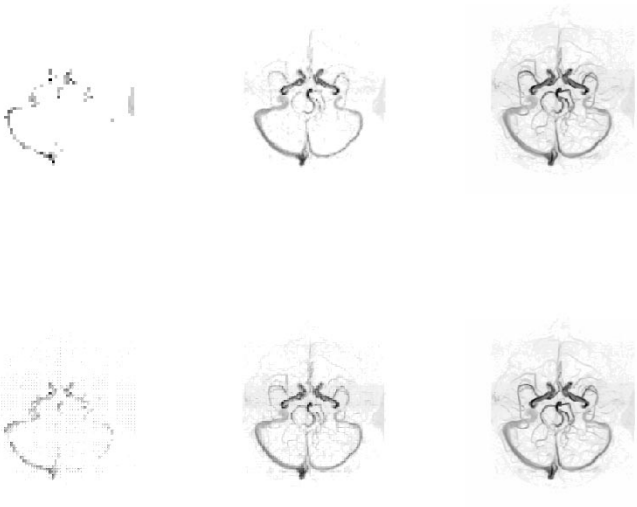

Fig. 3. MMIP reconstruction of the MRA data set from a 2-level morphological adjunction pyramid. First row: flat structuring function; second row: nonflat structuring function (support size $2 \times 2 \times 2$). From left to right: approximations on level 2, level 1 and level 0 (original).

V. Discussion

In this paper, we have described a multiresolution extension to MIP volume rendering, allowing progressive refinement and perfect reconstruction. The method is based on a particular type of nonlinear pyramids, the *morphological adjunction* pyramid. The pyramidal analysis and synthesis operators are composed of morphological 3-D erosion and dilation, combined with dyadic downsampling for analysis and dyadic upsampling for synthesis.

The proposed multiresolution MIP (MMIP) algorithm can be summarized as follows. In a preprocessing step, an L-level 3-D morphological adjunction pyramid of the volume data is computed, resulting in a sequence $d_0, d_1, \ldots, d_{L-1}, f_L$, where $\{d_j\}$ are detail signals and f_L is a signal at the coarsest level. After choosing a view direction, actual MIP volume rendering takes place by first computing a low resolution approximation from the coarsest signal f_L, and then refining the image progressively by taking the detail signals d_k, $k = L-1, \ldots, 0$ into account. We have shown that the operations of maximum projection and pyramidal 3-D synthesis can be interchanged. As a consequence, maximum projection of the coarser signals can be computed first, followed by a 2-D pyramid synthesis operation and a 2-D closing, resulting in a computationally efficient algorithm. The algorithm is very similar to wavelet splatting [5], [9], the main differences being that i) linear summation of voxel values is replaced by maximum computation and ii) linear wavelet filters are replaced by morphological filters (dilation and erosion). If the volume data are of integer type (bytes or shorts, depending on the dynamic range), then in contrast to the case of wavelet-based volume rendering, no floating point computations are required. An efficient implementation was obtained by using an object order voxel projection scheme [13]. Further memory reduction is possible by compression domain rendering, just as in the case of wavelet splatting [6].

There are a number of issues to be addressed in future work. First, adjunction pyramids quickly remove details of the data at higher approximation levels. To improve on this, other operators instead of erosions can be used for the analysis phase, such as openings. This implies however, that the pyramid is no longer an adjunction pyramid, and in particular, that the representation formula (7) no longer holds. To maintain an acceptable level of efficiency, we still require that the synthesis operator $\psi^{\downarrow}$ is a dilation, so that it commutes with maxima. Examples of such morphological pyramids are known, for example the so-called Sun–Maragos pyramid [25], see also [11]. Second, the special form of the subtraction operator in adjunction pyramids does have the disadvantage that the detail signals themselves are not "small" in regions where the structuring function does not match the data well. So, for better compression the use of ordinary addition and subtraction operators may be preferable. Again, this requires a different pyramid representation. Finally, we want to study the use of morphological wavelets [12], which have the advantage that they provide a nonredundant multiresolution representation, thus further saving memory.

REFERENCES

[1] R. A. Drebin, L. Carpenter, and P. Hanrahan, "Volume rendering," in *Computer Graphics (SIGGRAPH '88) Proc.*, vol. 22, 1988, pp. 65–74.

[2] S. Muraki, "Volume data and wavelet transforms," *IEEE Comput. Graph. Applicat.*, vol. 13, no. 4, pp. 50–56, 1993.

[3] R. Westermann and T. Ertl, "A multiscale approach to integrated volume segmentation and rendering," in *Proc. Eurographics'97*, D. Fellner and L. Szirmay-Kalos, Eds. Vienna, 1997, vol. 16, no. 3, pp. C-117–C-127.

[4] R. Grosso and T. Ertl, "Biorthogonal wavelet filters for frequency domain volume rendering," in *Proc. Visualization in Scientific Computing '95*, R. Scateni, J. van Wijk, and P. Zanarini, Eds., 1995, pp. 81–95.

[5] L. Lippert and M. H. Gross, "Fast wavelet based volume rendering by accumulation of transparent texture maps," *Comput. Graph. Forum*, vol. 14, no. 3, pp. 431–443, 1995.

[6] L. Lippert, M. H. Gross, and C. Kurmann, "Compression domain volume rendering for distributed environments," in *Proc. Eurographics'97*, 1997, pp. 95–107.

[7] L. A. Westover, "Footprint evaluation for volume rendering," *Comput. Graph.*, vol. 24, no. 4, pp. 367–376, 1990.

[8] J. B. T. M. Roerdink and M. A. Westenberg, "Wavelet-based volume visualization," *Nieuw Archief voor Wiskunde*, ser. Fourth, vol. 17, no. 2, pp. 149–158, July 1999.

[9] M. A. Westenberg and J. B. T. M. Roerdink, "Frequency domain volume rendering by the wavelet X-ray transform," *IEEE Trans. Image Processing*, vol. 9, pp. 1249–1261, July 2000.

[10] T. Malzbender, "Fourier volume rendering," *ACM Trans. Graphics*, vol. 12, no. 3, pp. 233–250, 1993.

[11] J. Goutsias and H. J. A. M. Heijmans, "Multiresolution signal decomposition schemes. Part 1: Linear and morphological pyramids," *IEEE Trans. Image Processing*, vol. 9, pp. 1862–1876, Nov. 2000.

[12] H. J. A. M. Heijmans and J. Goutsias, "Multiresolution signal decomposition schemes. Part 2: Morphological wavelets," *IEEE Trans. Image Processing*, vol. 9, pp. 1897–1913, Nov. 2000.

[13] L. Mroz, A. König, and E. Gröller, "Maximum intensity projection at warp speed," *Comput. Graph.*, vol. 24, pp. 343–352, 2000.

[14] K. J. Zuiderveld, A. H. J. Koning, and M. A. Viergever, "Techniques for speeding up high-quality perspective maximum intensity projection," *Pattern Recognit. Lett.*, vol. 15, pp. 507–517, 1994.

[15] W. Cai and G. Sakas, "Maximum intensity projection using splatting in sheared object space," in *Comput. Graph. Forum (Proc. Eurographics'98)*, vol. 17, 1998, pp. C113–124.

[16] J. Serra, *Image Analysis and Mathematical Morphology*. New York: Academic, 1982.

[17] H. J. A. M. Heijmans, *Morphological Image Operators, Advances in Electronics and Electron Physics, Supplement*. New York: Academic, 1994, vol. 25.

[18] J. B. T. M. Roerdink and A. Meijster, "The watershed transform: Definitions, algorithms, and parallellization strategies," *Fundamenta Informaticae*, vol. 41, pp. 187–228, 2000.

[19] C. Lürig and T. Ertl, "Hierarchical volume analysis and visualization based on morphological operators," in *Proc. IEEE Visualization '98*, 1998, pp. 335–341.

[20] J. B. T. M. Roerdink and G. S. M. Blaauwgeers, "Visualization of Minkowski operations by computer graphics techniques," in *Mathematical Morphology and Its Applications to Image Processing*, J. Serra and P. Soille, Eds. Dordrecht, The Netherlands: Kluwer, 1994, pp. 289–296.

[21] J. B. T. M. Roerdink, "Multiresolution maximum intensity volume rendering by morphological pyramids," in *Data Visualization 2001. Proc. Joint Eurographics—IEEE TCVG Symposium on Visualization, May 28–30, 2001, Ascona, Switzerland*, D. Ebert, J. M. Favre, and R. Peikert, Eds. New York: Springer, 2001, pp. 45–54.

[22] H. J. A. M. Heijmans and A. Toet, "Morphological sampling," *Comp. Vis. Graph. Image Proc.: Image Understanding*, vol. 54, pp. 384–400, 1991.

[23] P. J. Burt and E. H. Adelson, "The Laplacian pyramid as a compact image code," *IEEE Trans. Commun.*, vol. 31, pp. 532–540, 1983.

[24] M. A. Westenberg and J. B. T. M. Roerdink, "X-ray volume rendering through two-stage splatting," *Mach. Graph. Vis.*, vol. 9, no. 1/2, pp. 307–314, 2000.

[25] F.-K. Sun and P. Maragos, "Experiments on image compression using morphological piramids," *Proc. SPIE*, vol. 1199, pp. 1303–1312, 1989.

Jos B. T. M. Roerdink (M'95) received the M.Sc. degree in theoretical physics in 1979 from the University of Nijmegen, The Netherlands. He received the Ph.D. degree in 1983 from the University of Utrecht, The Netherlands.

From 1983 to 1985, he was a Postdoctoral Fellow at the University of California, San Diego. From 1986 to 1992, he was with the Centre for Mathematics and Computer Science, Amsterdam, The Netherlands, where he worked on image processing and tomographic reconstruction. He was appointed Associate Professor (1992) and Full Professor (2003) at the Institute for Mathematics and Computing Science of the University of Groningen, where he currently holds a chair in scientific visualization and computer graphics. His current research interests include biomedical visualization, neuroimaging and bioinformatics.